system, using a typical fault-tolerance mechanism, automatically replays a non-idempotent action for specific data items in a distributed data set partition. However, exemplary embodiments for exactly-once semantics for streaming analytics in non-idempotent output operations are described below with references to FIGS. **1-4**.

[0015] Implementation of such exemplary embodiments may take a variety of forms, and exemplary implementation details are discussed subsequently with reference to the Figures.

[0016] FIG. **1** is a functional block diagram illustrating a distributed computing environment **100**, according to an exemplary embodiment. FIG. **1** provides only an illustration of one implementation and does not imply any limitations with regard to the environments in which different embodiments may be implemented. Many modifications of the depicted environment may be made by those skilled in the art without departing from the scope of the invention as recited by the claims. In some embodiments, the distributed computing environment **100** includes a network **106**, a driver **104**, which operates stream computing program **102**, one or more clients **108**, and one or more compute nodes **110**.

[0017] Network **106** interconnects driver **104**, one or more clients **108**, and one or more compute nodes **110**. In general, network **106** can be any combination of connections and protocols capable of supporting communication between driver **104**, one or more clients **108**, one or more compute nodes **110**, and stream computing program **102**. In some exemplary embodiments, network **106** can be a message bus. In an exemplary embodiment, stream computing program **102** implements network **106**, using a cluster of compute nodes, to scale to handle larger message rates. Network **106** can include wire cables, wireless communication links, fiber optic cables, routers, switches, firewalls, or any combination that can include wired, wireless, or fiber optic connections known by those skilled in the art.

[0018] In some exemplary embodiments, driver **104** hosts stream computing program **102**. In one exemplary embodiment, driver **104** can be any programmable electronic device or computing system capable of receiving and sending data, via network **106**, and performing computer-readable program instructions known by those skilled in the art. In an exemplary embodiment, driver **104** can be a central server. In some exemplary embodiments, driver **104** can include a storage database **300** (shown in FIG. **3**) for storing data including, but not limited to, EntityIDs, EventIDs, MinibatchIDs, PartitionIDs, sets of EventIDs, a Counters Table, a Streams Table, and an Events Table. In an exemplary embodiment, Storage Database **300** can also store minibatches and partitions. Storage Database **300** can be any programmable electronic device or computing system capable of receiving, storing, and sending files and data, and performing computer readable program instructions capable of communicating with driver **104**, one or more clients **108**, and one or more compute nodes **110**, via network **106**. In an exemplary embodiment, driver **104** can be a coordinator or orchestrator for the one or more compute nodes **110**. In some exemplary embodiments, driver **104** can be a cluster of compute nodes, in the distributed system, operating stream computing program **102**, via network **106**. In an exemplary embodiment, stream computing program **102** includes databases (not shown) that provide a service to external systems. In an exemplary embodiment, stream computing program **102** resides locally on driver **104**. In yet other exemplary embodiments, stream computing program **102** resides locally on one or more compute nodes **110**.

[0019] In some exemplary embodiments, driver **104** includes stream computing program **102** that utilizes components in order to detect fault-tolerance roll-backs in distributed dataset partitions and to determine which elements in the partition should not be processed to avoid replaying a non-idempotent action. For example, stream computing program **102** utilizes driver **104** configured to assign an eventid to a corresponding entityid, to determine a minibatchid and a partitionid for a partition, to determine whether the partition was previously processed, to generate a new minibatchid and a new partitionid for a new partition based upon determining the partition was not previously processed, and to determine whether a record was previously processed based upon determining the partition was previously processed; and one or more compute nodes **110** configured to process the record of the partition based upon the driver **104** determining the record was not previously processed.

[0020] In some exemplary embodiments, stream computing program **102** operates on a central server, such as driver **104**, and can be utilized by one or more clients **108** and by one or more compute nodes **110** via a mobile application downloaded from the central server or a third-party application store, and executed on the one or more clients **108** and one or more compute nodes **110**. In some exemplary embodiments, stream computing program **102**, utilizing driver **104**, can route a partition of a minibatch **201** (shown in FIG. **2**) to a compute node **110** using a partitioning scheme. In other exemplary embodiments, stream computing program **102** routes one or more partitions of a minibatch **201** to a compute node **110**. In yet other exemplary embodiments, stream computing program **102**, utilizing driver **104**, coordinates the processing at one or more compute nodes **110**. In an exemplary embodiment, driver **104**, operating stream computing program **102**, runs on a specific compute node **110** and starts tasks that are distributed across one or more compute nodes **110**.

[0021] In some exemplary embodiments, one or more compute nodes **110** can provide a service that can be accessed by one or more clients **108**. In an exemplary embodiment, messages from a specific user or entity of a client **108** can be processed by any of one or more compute nodes **110**.

[0022] In some exemplary embodiments, a client **108** is an agent to driver **104** and can be for example, a desktop computer, a laptop computer, a smart phone, or any other electronic device or computing system, known by those skilled in the art, capable of communicating with driver **104** through the network **106**. For example, client **108** may be a laptop computer capable of accessing stream computing program **102** through a network, such as network **106**, and providing messages. In other exemplary embodiments, client **108** can be any suitable types of mobile devices capable of running mobile applications or a mobile operating system. In yet another exemplary embodiment, client **108** can be an intermediary, such as a website, between an end user and stream computing program **102**.

[0023] In some exemplary embodiments, a compute node **110** can be any programmable electronic device or computing system capable of receiving and sending data, via network **106**, and performing computer-readable program instructions known by those skilled in the art. In some